Summer Program FGV/EMAp 2019

Introduction to Machine Learning with Python

Regression Models

Prof. Luis Gustavo Nonato

University of São Paulo - São Carlos - SP

Basic Concepts

Regression is the process of modeling the relation between *dependent* and *independent* variables.

Basic Concepts

Regression is the process of modeling the relation between *dependent* and *independent* variables.

Linear Regression assumes the model has the form:

$$y_i \approx f(\mathbf{x_i}) = \beta_0 + \sum_{j=1}^{d} x_{ij}\beta_j$$

Basic Concepts

Regression is the process of modeling the relation between *dependent* and *independent* variables.

Linear Regression assumes the model has the form:

$$y_i \approx f(\mathbf{x_i}) = \beta_0 + \sum_{j=1}^{d} x_{ij}\beta_j$$

Typically, we have training data $(\mathbf{x}_1, y_1) \ldots (\mathbf{x}_n, y_n)$ where each $\mathbf{x}_i = (x_{i1}, \ldots, x_{id})$ is a d-dimensional vector and y_i is a scalar.

Basic Concepts

Regression is the process of modeling the relation between *dependent* and *independent* variables.

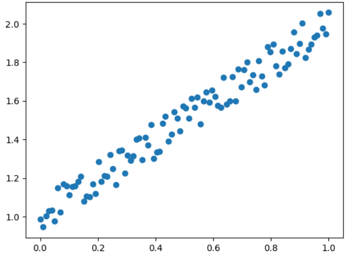
Linear Regression assumes the model has the form:

$$\boxed{y_i \approx f(\mathbf{x_i}) = \beta_0 + \sum_{j=1}^{d} x_{ij}\beta_j}$$

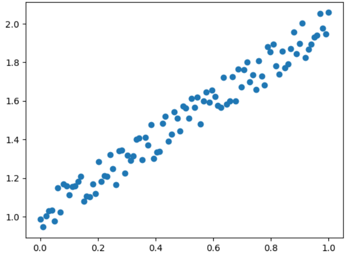
Typically, we have training data $(\mathbf{x}_1, y_1) \ldots (\mathbf{x}_n, y_n)$ where each $\mathbf{x}_i = (x_{i1}, \ldots, x_{id})$ is a d-dimensional vector and y_i is a scalar.

Properly estimating the parameters β_j is the main goal of a linear regression.

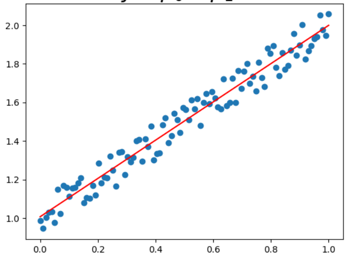
Basic Concepts



Basic Concepts



$$y = \beta_0 + \beta_1 x$$



Residual Sum of Squares

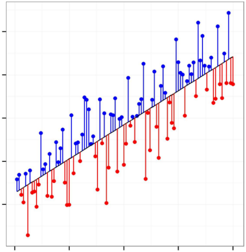
The **least squares** method computes the parameters $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_d)^\top$ so as to minimize the **residual sum of squares**

$$RSS(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2 = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{d} x_{ij}\beta_j \right)^2$$

Residual Sum of Squares

The **least squares** method computes the parameters $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_d)^\top$ so as to minimize the **residual sum of squares**

$$RSS(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2 = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{d} x_{ij}\beta_j \right)^2$$



Residual Sum of Squares

The **least squares** method computes the parameters $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_d)^\top$ so as to minimize the **residual sum of squares**

$$RSS(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2 = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{d} x_{ij}\beta_j \right)^2$$

$$\begin{bmatrix} y1 \\ y2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1d} \\ 1 & x_{21} & \cdots & x_{2d} \\ & & \vdots & \\ 1 & x_{n1} & \vdots & x_{nd} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_d \end{bmatrix}$$

Residual Sum of Squares

The **least squares** method computes the parameters $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_d)^\top$ so as to minimize the **residual sum of squares**

$$RSS(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2 = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{d} x_{ij}\beta_j \right)^2$$

$$\begin{bmatrix} y1 \\ y2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1d} \\ 1 & x_{21} & \cdots & x_{2d} \\ & & \vdots & \\ 1 & x_{n1} & \vdots & x_{nd} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_d \end{bmatrix}$$

$$RSS(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

Residual Sum of Squares

Differentiating $RSS(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$ and setting the derivative to zero

$$\frac{\partial RSS(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = 0 \Rightarrow \mathbf{X}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = 0$$

Residual Sum of Squares

Differentiating $RSS(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$ and setting the derivative to zero

$$\frac{\partial RSS(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = 0 \Rightarrow \mathbf{X}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = 0$$

Rearranging the terms we have

$$\boxed{\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}}$$

(Least Squares Estimates)

Residual Sum of Squares

Differentiating $RSS(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$ and setting the derivative to zero

$$\frac{\partial RSS(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = 0 \Rightarrow \mathbf{X}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = 0$$

Rearranging the terms we have

$$\boxed{\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}}$$

(Least Squares Estimates)

In practice $\hat{\boldsymbol{\beta}}$ is computed solving the system $(\mathbf{X}^\top\mathbf{X})\boldsymbol{\beta} = \mathbf{X}^\top\mathbf{y}$ (using QR factorization).

Biased $\times$ Unbiased Estimates

An important mathematical result attests that the least squares estimates of $\boldsymbol{\beta}$ have the smallest variance among all linear unbiased estimates.

Biased × Unbiased Estimates

An important mathematical result attests that the least squares estimates of $\boldsymbol{\beta}$ have the smallest variance among all linear unbiased estimates.

Considering the *mean square error* and assuming $\hat{\boldsymbol{\beta}}$ an estimate (not necessarily the least squares) we have

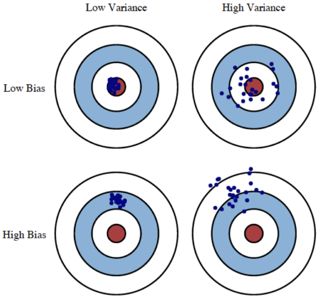
$$MSE(\hat{\boldsymbol{\beta}}) = E(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^2 = \underbrace{Var(\hat{\boldsymbol{\beta}})}_{variance} + \underbrace{\left[E(\hat{\boldsymbol{\beta}}) - \boldsymbol{\beta}\right]^2}_{\substack{bias \\ (=0 \text{ for unbiased})}}$$

Biased × Unbiased Estimates

An important mathematical result attests that the least squares estimates of $\boldsymbol{\beta}$ have the smallest variance among all linear unbiased estimates.

Considering the *mean square error* and assuming $\hat{\boldsymbol{\beta}}$ an estimate (not necessarily the least squares) we have

$$MSE(\hat{\boldsymbol{\beta}}) = E(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^2 = \underbrace{Var(\hat{\boldsymbol{\beta}})}_{variance} + \underbrace{\left[E(\hat{\boldsymbol{\beta}}) - \boldsymbol{\beta}\right]^2}_{\substack{bias \\ (=0\,for\,unbiased)}}$$



Biased $\times$ Unbiased Estimates

The previous discussion raises the following question:

Biased $\times$ Unbiased Estimates

The previous discussion raises the following question:

Is it possible to find a biased estimator with smaller variance?

Biased $\times$ Unbiased Estimates

The previous discussion raises the following question:

Is it possible to find a biased estimator with smaller variance?

The answer is YES !!

Biased $\times$ Unbiased Estimates

The previous discussion raises the following question:

Is it possible to find a biased estimator with smaller variance?

The answer is YES !!

There are several ways to obtain such biased estimates !!

– *subset selection*

– *regularized optimization schemes*

Subset Selection

Subset selection aims to:

Subset Selection

Subset selection aims to:

- introducing a bit of bias in favor of reducing variance;

Subset Selection

Subset selection aims to:

- introducing a bit of bias in favor of reducing variance;
- make the identification and interpretation of relevant attributes an easier task.

Subset Selection

Subset selection aims to:

- introducing a bit of bias in favor of reducing variance;
- make the identification and interpretation of relevant attributes an easier task.

$$y \approx f(x) = \beta_0 + x_1\beta_1 + x_2\beta_2 + x_3\beta_3 + \cdots + x_d\beta_d$$

Subset Selection

Subset selection aims to:

- introducing a bit of bias in favor of reducing variance;
- make the identification and interpretation of relevant attributes an easier task.

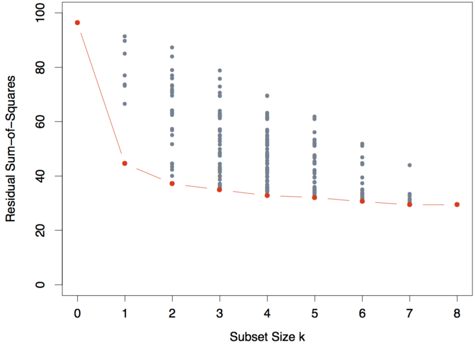
$$y \approx f(x) = \underset{\uparrow}{\beta_0} + x_1\beta_1 + \underset{\uparrow}{x_2\beta_2} + x_3\beta_3 + \cdots + \underset{\uparrow}{x_d\beta_d}$$

Best-Subset Selection

Best "RSS $\times$ # Variables" trade-off among all possible subsets.

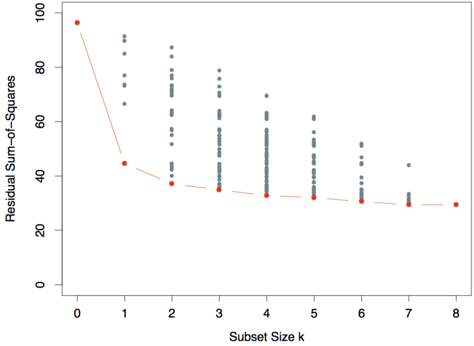
Best-Subset Selection

Best "RSS $\times$ # Variables" trade-off among all possible subsets.



Best-Subset Selection

Best "RSS $\times$ # Variables" trade-off among all possible subsets.



Computationally unfeasible for large values of d !!

Forward- and Backward-Stepwise

Greedy-like alternatives to reduce the computational burden !!

Forward- and Backward-Stepwise

Greedy-like alternatives to reduce the computational burden !!

Forward Selection

- Starts with β_0 (the intercept) and then sequentially adds into the model the predictor that most improves the fit.

Forward- and Backward-Stepwise

Greedy-like alternatives to reduce the computational burden !!

Forward Selection

- Starts with β_0 (the intercept) and then sequentially adds into the model the predictor that most improves the fit.

Backward Selection

- Starts with the full model and then sequentially deletes the predictor that has the least impact on the fit.

Forward- and Backward-Stepwise

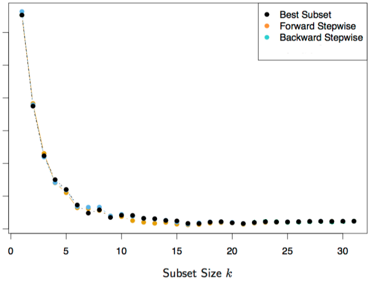
Greedy-like alternatives to reduce the computational burden !!

Forward Selection

- Starts with β_0 (the intercept) and then sequentially adds into the model the predictor that most improves the fit.

Backward Selection

- Starts with the full model and then sequentially deletes the predictor that has the least impact on the fit.



Shrinkage Methods

Similarly to subset selection, *shrinkage methods* deal with model parameters in order to improve accuracy while making attribute interpretation more manageable.

Shrinkage Methods

Similarly to subset selection, *shrinkage methods* deal with model parameters in order to improve accuracy while making attribute interpretation more manageable.

In contrast to the discrete nature of subset selection, shrinkage methods are continuous, resulting in lower variance parameter estimation.

Shrinkage Methods

Similarly to subset selection, *shrinkage methods* deal with model parameters in order to improve accuracy while making attribute interpretation more manageable.

In contrast to the discrete nature of subset selection, shrinkage methods are continuous, resulting in lower variance parameter estimation.

The idea is play with the full model, but imposing penalties to the parameters so as to shrink them to zero.

Ridge Regression

$$\hat{\boldsymbol{\beta}} = \operatorname{argmin}_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{d} x_{ij}\beta_j)^2 + \ s \sum_{j=1}^{d} \beta_j^2 \ \right\}$$

Ridge Regression

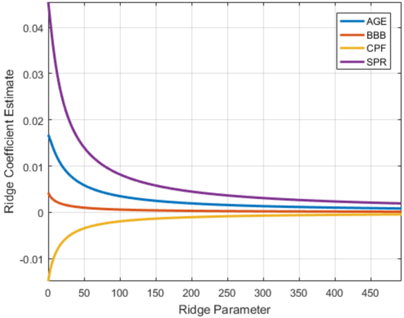
$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{d} x_{ij}\beta_j)^2 + \boxed{s \sum_{j=1}^{d} \beta_j^2} \right\}$$

The right most term is the *regularization term* and it forces the coefficients β_j shrink to zero.

Ridge Regression

$$\hat{\boldsymbol{\beta}} = \operatorname{argmin}_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{d} x_{ij}\beta_j)^2 + \boxed{s \sum_{j=1}^{d} \beta_j^2} \right\}$$

The right most term is the *regularization term* and it forces the coefficients β_j shrink to zero.



Ridge Regression

The intercept β_0 is not part of the regularization term.

Ridge Regression

The intercept β_0 is not part of the regularization term.

In practice we can remove the intercept by centralizing the data $x_{ij} - \overline{x}_j$ and making $\beta_0 = \frac{1}{n}\sum_i^n y_i$.

Ridge Regression

The intercept β_0 is not part of the regularization term.

In practice we can remove the intercept by centralizing the data $x_{ij} - \overline{x}_j$ and making $\beta_0 = \frac{1}{n}\sum_i^n y_i$.

Without β_0 we can write the residual sum of squares as:

$$RSS(\boldsymbol{\beta}, s) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + s\boldsymbol{\beta}^\top\boldsymbol{\beta}$$

Ridge Regression

The intercept β_0 is not part of the regularization term.

In practice we can remove the intercept by centralizing the data $x_{ij} - \overline{x}_j$ and making $\beta_0 = \frac{1}{n}\sum_i^n y_i$.

Without β_0 we can write the residual sum of squares as:

$$RSS(\boldsymbol{\beta}, s) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + s\boldsymbol{\beta}^\top \boldsymbol{\beta}$$

Differentiating w.r.t. $\boldsymbol{\beta}$ and setting the derivative to zero we get

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X} + s\mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}$$

Ridge Regression

What the regularization is indeed doing?

Ridge Regression

What the regularization is indeed doing?

Knowing that $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ and $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X} + s\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$

Ridge Regression

What the regularization is indeed doing?

Knowing that $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ and $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X} + s\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$

$$\mathbf{y} \sim \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}^\top\mathbf{X} + s\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y} \tag{1}$$

$$= \sum_{j=1}^{p} \mathbf{u}_j \, \frac{d_j^2}{d_j^2 + s} \, \mathbf{u}_j^\top \mathbf{y} \tag{2}$$

Ridge Regression

What the regularization is indeed doing?

Knowing that $\mathbf{X} = \mathbf{UDV}^\top$ and $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X} + s\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$

$$\mathbf{y} \sim \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}^\top\mathbf{X} + s\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y} \tag{1}$$

$$= \sum_{j=1}^{p} \mathbf{u}_j \boxed{\frac{d_j^2}{d_j^2 + s}} \mathbf{u}_j^\top \mathbf{y}, \qquad \frac{d_j^2}{d_j^2 + s} \leq 1 \tag{2}$$

Ridge Regression

What the regularization is indeed doing?

Knowing that $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ and $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X} + s\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$

$$\mathbf{y} \sim \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}^\top\mathbf{X} + s\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y} \tag{1}$$

$$= \sum_{j=1}^{p} \mathbf{u}_j \boxed{\frac{d_j^2}{d_j^2 + s}} \mathbf{u}_j^\top \mathbf{y}, \qquad \frac{d_j^2}{d_j^2 + s} \leq 1 \tag{2}$$

The smaller d_j and larger s the closer to zero $\frac{d_j^2}{d_j^2+s}$ is.

Ridge Regression

What the regularization is indeed doing?

Knowing that $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ and $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X} + s\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$

$$\mathbf{y} \sim \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}^\top\mathbf{X} + s\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y} \tag{1}$$

$$= \sum_{j=1}^{p} \mathbf{u}_j \boxed{\frac{d_j^2}{d_j^2 + s}} \mathbf{u}_j^\top \mathbf{y}, \qquad \frac{d_j^2}{d_j^2 + s} \leq 1 \tag{2}$$

The smaller d_j and larger s the closer to zero $\frac{d_j^2}{d_j^2+s}$ is.
Since small d_j are related to noise (remember PCA lesson !), rigde regression is making a "soft" selection of the main components, removing noise and writing data back in the original coordinate system.

Lasso

Lasso is similar to ridge regression,

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{d} x_{ij}\beta_j)^2 + \ s \sum_{j=1}^{d} |\beta_j| \ \right\}$$

Lasso

Lasso is similar to ridge regression, with a subtle but important difference in the regularization term.

$$\hat{\boldsymbol{\beta}} = \operatorname{argmin}_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{d} x_{ij}\beta_j)^2 + \boxed{s \sum_{j=1}^{d} |\beta_j|} \right\}$$

Lasso

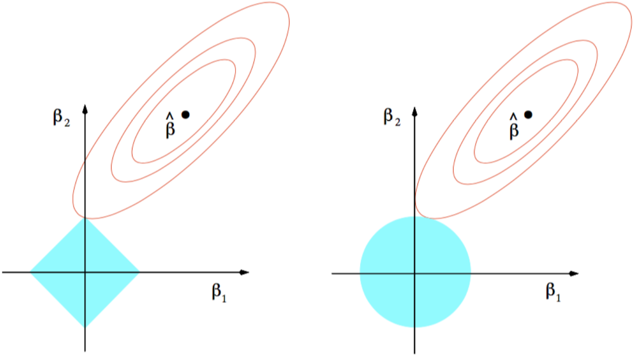
Lasso is similar to ridge regression, with a subtle but important difference in the regularization term.

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{d} x_{ij}\beta_j)^2 + \boxed{s \sum_{j=1}^{d} |\beta_j|} \right\}$$

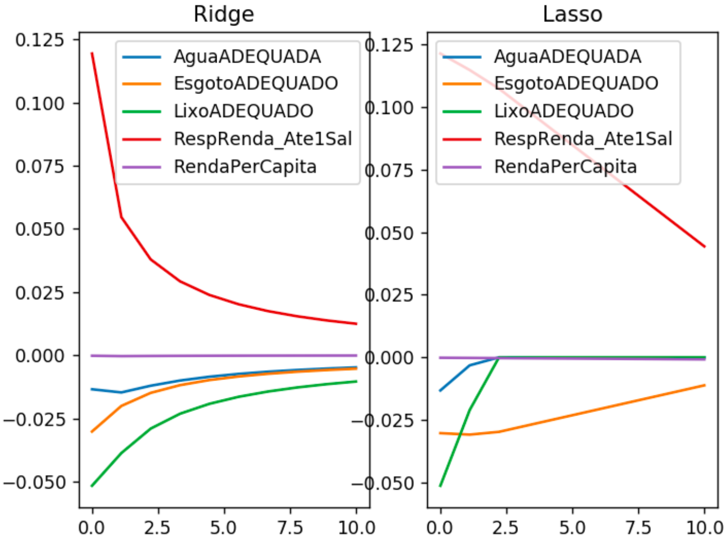
Unfortunately there is no closed form expression for the solution of the Lasso regression, thus a optimization procedure has to be employed to compute $\hat{\boldsymbol{\beta}}$.

Lasso

An import aspect of Lasso is that, tuning s properly, non-relevant parameters are quickly truncated to zero.



Lasso x Rigde



Cross-Validation

K-fold approach:

D1	D2	D3	D4	D5
Train	Train	Validation	Train	Train

Given a model M and a K-fold of a data set D

- for $k = 1, \ldots, K$
  - Consider the training set $D^{(-k)} = D/D_k$
  - Learn M from $D^{(-k)}$
  - $e_k(M) = \sum_{i \in D_k} (y_i - \hat{y}_i^{(-k)})^2$
- $CV(M) = \frac{1}{n} \sum_{k=1}^{K} e_k(M)$

Cross-Validation

K-fold approach:

D1	D2	D3	D4	D5
Train	Train	Validation	Train	Train

Given a model M and a K-fold of a data set D

- for $k = 1, \ldots, K$
  - Consider the training set $D^{(-k)} = D/D_k$
  - Learn M from $D^{(-k)}$
  - $e_k(M) = \sum_{i \in D_k} (y_i - \hat{y}_i^{(-k)})^2$
- $CV(M) = \frac{1}{n} \sum_{k=1}^{K} e_k(M)$

When $K = n$ the K-fold is called *leave-one-out cross-validation*.

Cross-Validation

Model assessment: having chosen a model, estimating its prediction error on new data.

Cross-Validation

Model assessment: having chosen a model, estimating its prediction error on new data.

K-fold can be used to assess the quality of a particular model.